Common Consistency Requirements for Data Grids, Digital Libraries, and Persistent Archives

Working Draft, Version 1.0

Status of this Memo: This memo is a draft for technical recommendations. Distribution is unlimited.

Author(s)	Institution(s)
Reagan W. Moore	San Diego Supercomputer Center University of California, San Diego 9500 Gilman Drive, MC-0505 La Jolla, CA 92093-0505 moore@sdsc.edu

Abstract

The goal of the Global Grid Forum is to develop a common set of services for access to distributed resources. Grid services rely upon the use of logical name spaces to provide global identifiers for the registration of resources, users, applications, and digital entities. A logical name space is a location independent naming convention. Grid services are implemented as middleware that manages the distributed state information needed to execute the service operations. The distributed state information is stored as a mapping of the service attributes to the logical name space. Consistency requirements can be described as constraints on the mappings. We will look at the types of constraints implemented by grid services, the impact of compositing services on the preservation of the constraints, and the use of ontologies to organize constraints on mappings within grids, digital libraries, and persistent archives.

Table of Contents

1   Introduction

Multiple management systems are being developed to organize digital entities for use by communities. Examples include digital libraries [15], which focus on publication and discovery mechanisms, persistent archives [12], which focus on preservation and technology evolution management, and data grids [13], which focus on interoperability across distributed resources. Each system provides a logical name space for referencing digital entities. Each system provides services that map distributed state information to the logical name space. Each system manages consistency requirements that define how the state information from different services can be integrated.   In this paper, we look at the characterization of consistency requirements, and build a model that allows consistency requirements to be integrated across multiple types of management environments. There are multiple usage scenarios that require the characterization and application of constraints. We will illustrate the management of consistency requirements for access to distributed resources that are manipulated by multiple services within a grid, for federation of independent virtual organizations, for integration of knowledge management systems on top of data grids, and for management of digital ontologies that describe the relationships present within digital entities.

2   Logical Name Spaces

Grids are used to access geographically distributed resources that reside in different administration domains, and that are labeled by different local naming conventions. The resources include compute platforms, storage systems, digital entities, applications and users. Each type of physical resource is typically named using a different convention. In addition, the naming conventions vary across the administration domains that control access to a given physical resource. An environment that is based on different naming conventions at each geographic site is extremely difficult to use. Grids implement registries for managing uniform identifiers for each type of resource. Compute platforms and storage systems are described in a resource description service (Globus Metadata Directory Service [5]), digital entities are named by a data grid (Globus Replica Catalog [5] or Storage Resource Broker metadata catalog [9]), applications are named by a workflow management system (Chimera application repository [4]), and users are described in an authentication system (GSI Certificate authority [7]).

Grids manage these heterogeneous naming conventions by creating logical name spaces that are location independent. The logical name space is used to create a global persistent identifier that uniquely names the physical resources across the sites. A physical resource name is mapped to a logical resource name, and the mapping is maintained in a registry. The mappings may be simple, with a single physical resource represented by a single logical resource name. The mappings may be more complex, with the logical resource name used to represent a set of physical resources.

The logical name space defines the identities upon which grid services operate. This leads to the concept that grids operate in logical name spaces. Operations are defined on the logical names, not the physical names. Each service defines attributes that are used to describe their distributed state information, which are mapped to the logical name space. Grids manage the distributed state information in the registries. This leads to the definition of middleware as a software system that manages distributed state information for services [2].

Digital libraries also implement a logical name space. The digital entities that are registered within a digital library are not restricted to files, but may include URLs, SQL commands, and collections of data and metadata. The services that digital libraries provide are typically associated with discovery on descriptive metadata. The descriptive metadata are managed as attributes in a collection and represent a mapping of semantic identifiers to the logical name space. One can build a data grid that also provides support for mapping descriptive metadata to the logical name space. One can build a digital library that accesses data distributed across the web. The distinction between data grids and digital libraries, from the perspective of the need to manage mappings on a logical name space, is disappearing.

Persistent archives also use a logical name space to support global identifiers that are held invariant over time. Persistent archives map authenticity metadata to the logical name space to manage technology evolution, and to support assertions of authenticity [8,16]. Since persistent archives must guarantee continued survival of digital entities in the presence of disasters, persistent archives also manage replication of digital entities across multiple sites. Persistent archives are being implemented from data grids [11]. The Persistent Archive Working Group of the Global Grid Forum [6] is examining the minimal set of grid capabilities that are needed to implement a persistent archive. In the process, this is defining requirements on data grids such that they can be used to create persistent archives. Given that mappings on the logical name space can be used to implement digital libraries and persistent archives, we will examine the consistency constraints that are emerging from the data grid community.

3 Mappings on logical name spaces

A logical name space is used to create global, persistent, location independent identifiers. A mapping is imposed on the logical name space to associate physical resource names with logical resource names. There are wide variety of mappings that can be applied to the logical name space:

- Spatial mapping – Map from the logical name to one or more physical names. An example is the association of a list of physical resource names with a single logical resource name. Writing to the logical resource name would result in a copy of the file being created on each physical resource. The distributed state information includes the location of each replica, the name of the physical file, the access protocol used to interact with the storage repository, the time of creation, the size of the file on that system, etc.

0 Structural mapping – Map from a logical file name to a location in a container. The management of the logical file is then subsumed within the management of the container. Operations on the physical file require manipulations on the container, and the extraction of the physical file from the container. Thus the container might be replicated, or migrated to an archive. Retrieving an individual file would require identifying the appropriate container, caching the container to disk, and extracting the requested file. The distributed state information would include the name of the container in which the file is located, the offset within the container, etc.

- Temporal mapping – Map from a logical file name to time-based snapshots. This is equivalent to the creation of versions of a file. The distributed state information would include the timestamp associated with the version, the location of the version, the name of the physical file that holds the version, the access protocol used to interact with the storage repository, etc.

- Procedural mapping – Map from a logical file name to versions of the physical file that are stored in different encoding formats. Instead of bit-for-bit copies, a replica is made

using an alternate encoding format. The distributed state information includes a mime type for the encoding format, the location of the semantic equivalent file, the name of the physical file used to hold the alternate encoding format, the access protocol used to interact with the storage repository, the time of creation, the size of the file, etc.

- Semantic mapping – Map from the logical file name to descriptive metadata that can be used to support discovery. A typical application is to associate user-defined attributes with a digital entity. This implies that every digital entity can be a collection of size one, with its own metadata.

- Control mapping – Map from a logical file name to a logical user name by assigning access controls on explicitly defined access roles. This is a triple mapping, in that allowed operations are defined by the permitted roles, the permission for a user is characterized as an access control relative to a role, and the access control is then associated with a logical file name. The result is that access controls are a property of the logical name, and are preserved when a file is moved between physical resources. This is an essential requirement for meeting the needs of the medical community for referencing data in distributed environments.

- Logical mapping – Impose a structure on the logical name space that can be used to organize the digital entities as either a directory/sub-directory hierarchy, or a collection/sub-collection hierarchy. Whichever is chosen, one can create soft links within the organization. References to a logical name may result in the traversal of a link to a second logical name, for which mappings exist to the distributed state information used for any of the above mappings. Distributed state information includes membership in a sub-collection, attributes associated with the collection, soft links between logical names, etc.

The mappings are maintained in registries. When multiple services are provided, each with distributed state information, a challenge is managing consistency of metadata across the services. A major question is whether the update of the distributed state information depends upon the order in which the services are invoked. Are the services both associative and commutative with respect to update of their distributed state information? Is there an expected order of execution when multiple services are invoked? Do operations by a particular service impact the allowed changes to the distributed state information that might be done by a subsequent service?

4 Consistency

Within a grid, one requirement of a service is that it either update the distributed state information to make it conform to the results of operations within the grid, or that it provide recovery mechanisms such that consistent distributed state information can be created. The distributed state information within a grid is very similar to the state information maintained within an operating system. It describes state properties that are needed to correctly identify and manipulate digital entities and resources. The grid does not have the luxury of crashing when the distributed metadata becomes inconsistent. Hence the desire either to execute recovery procedures, or to impose consistency requirements on the metadata update. We view consistency requirements as constraints on the mappings of the distributed state information onto the logical name space. Consistency constraints may be temporal, structural, logical, procedural, spatial, functional, etc. The management of consistency becomes even harder when the constraints are applied across multiple mappings. Again, there are many examples of consistency constraints:

- Temporal constraints – Within the SDSC Storage Resource Broker [3,17], the metadata that represents the distributed state information must be updated before the service is marked as complete. One can choose to impose soft state mechanisms such as in the Globus replica catalog, in which the distributed state information may be inconsistent, but a mechanism is provided to either recreate the state information by re-execution of the service or to execute a recovery procedure.

- Logical constraints – Within the SDSC SRB, all operations are preformed upon the logical name spaces. This means that the distributed state information is maintained independently of the physical resources. If the grid supports operations directly on the physical resources or physical file names without updating the distributed state iinformation, the update of the logical name space must be done out-of-path. Namely, the operation is completed and the distributed state information is updated by an independent service. If the update operation fails, a recovery mechanism must be provided, otherwise the system will remain logically inconsistent.

- Structural constraints – Within the SDSC SRB, the mapping of a logical resource name to a set of physical resource names may have multiple interpretations. Consider a logical resource name that represents a list of physical resource names. The following interpretations may be made by different services that operate on the logical resource name:
  - Replication – a write completes when copies exist on all of the physical resources
  - Load leveling – a write completes when a copy exists on the next resource in the list, and a counter is updated to point to the succeeding resource.
  - Fault tolerance – a write completes when copies exist on "k" of "n" of the physical resources.
  - Compound resource – a write completes when a copy exists on the disk cache associated with a tape-based system.

The constraints get even more involved when writing a file into a container that is replicated onto a logical resource that includes compound resources. In this case, the constraints include the ability to identify when the state is inconsistent (the updated file exists on only one of the physical resources), update mechanisms to force consistency across all of the resources (synchronization of copies), and control mechanisms to guarantee that further operations will not result in inconsistent state (locking mechanisms).

The imposition of consistency constraints within the SDSC SRB has been implemented as hard-coded software mechanisms. The addition of a new service that introduces a new mapping on the logical name space requires that the software code be changed. A major development effort is the creation of constraint-based collection management, in which the consistency constraints are implemented as relationships imposed on the logical name space mappings. A digital grid ontology is needed that describes the order in which the consistency requirements should be applied (to specify associative and commutative inconsistencies), the constraints that are imposed on the execution of other services, and the recovery mechanisms for reintroducing consistency within the distributed state information. The digital grid ontology specifies each constraint as a temporal or procedural relationship imposed on the mapping of distributed state information to the logical name space.

5   Grid Registries

Grid registries manage distributed state information for a virtual organization. Or conversely, a virtual organization is defined by the set of grid registries that manage the distributed state information. The virtual organization specifies the user logical name space (Certificate authority when using GSI authentication), the digital entity logical name space (replica catalog or SRB metadata catalog), the resource logical name space (Metdata Directory Service, or SRB metadata catalog), and the access logical name space (Community Authorization Server, or SRB metadata catalog). A virtual organization manages interactions between the logical name spaces to ensure consistency. For example, access controls correctly map from the user logical name space to the digital entity logical name space.

The scalability of grid registries is an implementation concern. The management of distributed state information may be accomplished through the use of LDAP directories or relational databases. Already within the grid environment, services operate on collections of millions to billions of digital entities (2MASS image archive [1], USNO-B proper motion catalog [18]). Grids are expected to scale in size. A corresponding requirement is the elimination of bottlenecks where all information must flow through a single resource. Hence the use of a central registry to manage a logical name space may impose both performance and fault tolerance limitations.

Grids work across distributed resources. Since the speed of light is slow compared to the performance of computers, grids must manage latency. This is especially true when dealing with large numbers of small objects. An object is small when its size is less than the product of the transmission bandwidth and the latency. A traditional way to handle latency is to use aggregation to minimize the number of message that must be sent. This includes aggregation of state information into a central repository. Grids must also manage consistency, updating the distributed state information on completion of each service. This is easier to implement when a central repository is used to manage the distributed state information.

There exists a fundamental tension in the implementation of grid registries, between the elimination of single points of failure, the management of latency, and the management of consistency. There are several approaches towards resolving these tensions:

- Keep state information local, such that all state information is implemented in a local registry. This avoids the issue of latency management, provided there is a way to replicate the distributed state information across the local repositories. The management of consistency becomes very difficult if every service must send updates to every registry. To avoid the overhead, the local registries are updated periodically.

- Keep state information in a central repository, and rely upon database technology to replicate the state information across backup sites. If the initial repository is down, the system accesses a back-up site. The performance is limited to the capability of the resources on which the central repository is built.

- Manage the registry as a soft state mechanism. A local registry is used to minimize latency, a hierarchy is provided to access non-local information, and the hierarchy is periodically updated. If the desired state information is not available at the next level of the hierarchy, then the service is either re-executed to re-generate a replica, or the request is forwarded to the remote resources for resolution. This approach assumes the state information is being managed within a virtual organization that has implemented uniform consistency constraints.

- Manage the registry as a federation of peers. This requires the creation of soft links between the registries that may belong to different virtual organizations. The challenge is that the consistency constraints may be different in each virtual organization. They may use different certificate authorities, different community authorization servers, may impose different temporal update constraints, etc. One example is the management of access control lists on a logical entity that is registered into a second virtual organization. Do the access controls follow the digital entity, implying that the second virtual organization must be able to check access restrictions with the original virtual organization? Do the descriptive metadata associated with a digital entity go with the soft link as a time-stamped snapshot, or are the metadata updated whenever a change is made within the original collection? Is the distributed state information about replicas provided to the second virtual organization? Depending upon the policies in force between the virtual organizations, the consistency mechanisms may be implemented as update "push" or demand "pull".

The generalization of a grid registry is a constraint-based collection that explicitly specifies the consistency constraints that are maintained on the distributed state information. The consistency constraints can be characterized as the set of temporal, structural, and procedural relationships that are maintained on the mapping. The ability to specify these constraints, such as the registration of a digital entity from another virtual organization, needs to be a property that is associated with the logical name space. A constraint-based collection provides support for not only the distributed state information, but also the constraints that are imposed on updates of the distributed state information.

6   Constraint-based Collections

The ability to apply relationships to collections is an emerging requirement of data grids. We need to be able to apply the relationships in evaluating consistency constraints across grid services. We also need to be able to organize the relationships into an ontology for deciding the order of application of services. When we attempt to federate logical name spaces across virtual organizations, we need to be able to compare ontologies, and determine which consistency constraints are compatible.

Ontologies are emerging within the grid communities for the description of workflows. The transformations between different states are organized as an ontology describing state transformations. Within the digital library community, a standard use of ontologies has been the development of logical relationships between semantic terms to support digital library crosswalks.

There are other examples of the desire to integrate ontologies onto data grids. The National Science Digital Library [14] is examining the integration of constraints on the mapping of digital entities into their logical name space. The constraints identify grade-relevance of curricula modules. A ontology is generated to identify scientific learning units as a function of grade level. The vocabulary as a function of grade level is specified in a second ontology. A mapping is applied to determine whether the vocabulary used within a curriculum module corresponds to both the grade level and the desired scientific learning unit. This is a constraint on the logical organization of the material within the collection, and should be applicable without having to rewrite the collection management software.

The persistent archive community is examining the management of technology evolution, including the ability to manipulate digital entities when encoding formats become obsolete. Archival processes are applied to create a representation of the digital entity that characterizes the internal relationships, such as the creation of structures from the bit

stream, the mapping of the structures to coordinate systems, the assignment of semantic labels to the structures, the assignment of physical units to semantic labels, the assignment of temporal relationships, etc. The mappings can be organized into a digital ontology that represents the procedural order that is needed to transform bits into semantically labeled components [10]. The digital ontology is then used as the characterization of the digital entity. The digital ontology can be migrated onto new encoding standards for annotating relationships and semantic tags, while the original bit stream within the digital entity is preserved unchanged. The emulation program operates on the digital ontology, as a generic mapping from defined relationships to permitted operations. As new technology becomes available that allows additional operations on given relationships, the emulation environment becomes more sophisticated and richer. The persistent archive community, therefore, needs to mange not just distributed state information, but also the digital ontologies that characterize the information and knowledge content within digital entities. Grid ontologies are needed that characterize the collective properties of the digital ontologies. Operations on digital entities will be controlled by the set of relationships that are in common across the digital entities. Grid services would correspond to operations on these common relationships. The distributed state information then corresponds to mappings imposed on the logical name space for the manipulation of the common relationships within the digital ontologies. Grid services will be expressible as a grid ontology, specifying which operations will be performed on the underlying digital ontologies.

We propose identifying grid services as operations on relationships organized in ontologies. The grid ontology is the basic description of the consistency requirements needed for interoperation across virtual organizations.

7   Summary

The development of a common set of grid services for the manipulation of digital entities can be viewed as the creation of a hierarchy of ontologies. A digital ontology can be defined that describes the relationships present within a digital entity, and the order of application of the relationships that are needed to manipulate the digital entity. Operations on the digital entities correspond to operations on the relationships that have been organized within the digital ontology. Grid services correspond to operations on relationships that are present in common across collections of digital entities. Grid services manage distributed state information that is mapped to logical name spaces that are used to name digital entities as well as resources. The consistency constraints within grids can be expressed as a grid ontology that specifies the order of application of operations on the common digital ontology relationships. Finally, the integration of knowledge management environments on top of grids corresponds to the integration of additional sets of constraints on the grid logical name spaces, in this case to support organization for discovery and access.

8   Acknowledgements

9 Bibliography

- 2MASS, Two-Micron All Sky Survey, http://www.ipac.caltech.edu/2mass/

- Aiken, B., B. Carpenter, I. Foster, J. Mambretti, R. Moore, J. Strassner, B. Teitelbaum, Middleware Workshop, Northwestern University, Dec 1998, http://www-fp.mcs.anl.gov/middleware98/report.html,

- Baru, C., R, Moore, A. Rajasekar, M. Wan, "The SDSC Storage Resource Broker," Proc. CASCON'98 Conference, Nov.30-Dec.3, 1998, Toronto, Canada.

- The Chimera Virtual Data System, http://www-unix.griphyn.org/chimera/.

- Globus Toolkit™, http://www.globus.org/toolkit/.

- Grid Forum Persistent Archive Research Group, http://www.gridforum.org/6_DATA/persist.htm.

- GSI - Grid Security Infrastructure, http://www.globus.org/security/.

- Ludäscher, B., R. Marciano, R. Moore, "Towards Self-Validating Knowledge-Based Archives," pp. 9-16, RIDE-DM 2001.

- MCAT - "The Metadata Catalog", http://www.npaci.edu/DICE/SRB/mcat.html

- Moore, R., "The San Diego Project: Persistent Objects", Proceedings of the Workshop on XML as a Preservation Language, Urbino, Italy, October 2002.

- Moore, R., "Knowledge-based Persistent Archives," Proceedings of La Conservazione Dei Documenti Informatici Aspetti Organizzativi E Tecnici, in Rome, Italy, October, 2000.

- Moore, R., C. Baru, A. Rajasekar, B. Ludascher, R. Marciano, M. Wan, W. Schroeder, and A. Gupta, "Collection-Based Persistent Digital Archives – Parts 1& 2", D-Lib Magazine, April/March 2000, http://www.dlib.org/

- Moore, R., C. Baru, A. Rajasekar, R. Marciano, M. Wan, "Data Intensive Computing", in ``The Grid: Blueprint for a New Computing Infrastructure'', eds. I. Foster and C. Kesselman. Morgan Kaufmann, San Francisco, 1999.

- NSDL - National Science Digital Library, http://www.nsdl.org/.

- Rajasekar, A., R. Moore, B. Ludäscher, I. Zaslavsky, " The Grid Adventures: SDSC's Storage Resource Broker and Web Services in Digital Library Applications", 4th Russian Conference on Digital Libraries, Dubna, Russia, October 2002.

- Reference Model for an Open Archival Information System (OAIS). submitted as ISO draft, http://www.ccsds.org/documents/pdf/CCSDS-650.0-R-1.pdf, 1999.

- SRB - "The Storage Resource Broker Web Page, http://www.npaci.edu/DICE/SRB/

- Levine, S. E., A. S. Bosh, D. G. Monet, "The USNO-B Catalogue, and a Search for Candidate Occultation Stars", http://www.aas.org/publications/baas/v34n3/dps2002/238.htm

10 Glossary of Acronyms

•

11 Security Considerations

There are no currently identified security issues..Notices

11.1 Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director (see contacts information at GGF website).

11.2 Copyright Notice